

Selenium Tutorial

Contents

Web Elements.....	2
Locators.....	3
Xpath.....	3
CSS Selector.....	5
Wait (Test / Page Synchronization).....	6
Visibility of WebElement.....	9
Window Handles.....	9
Browser Navigation Commands.....	10
Actions Class (Keyboard and Mouse activities).....	10
o	
JavaScriptExecuter.....	14
Iframes.....	15
Model Window.....	16
Turn off notifications.....	16
Take Screen Shot.....	16
Advanced Selenium.....	17
AutoIT.....	17
Robot Class.....	17

Please Join in below link

Instagram: https://www.instagram.com/rba_infotech/

LinkedIn: <https://www.linkedin.com/company/rba-infotech/>

Facebook:

<https://www.facebook.com/people/RBA-Infotech/100043552406579/>

1. Web Elements

List of very important Elements:

1. Textbox
2. Button
3. Anchor Link - a href
4. Check Box
5. Radio button
6. Dropdown box / Select
7. Date Picker
8. Text area
9. Image, Image link and Image button

List of few HTML Web Elements:

<https://www.w3schools.com/tags/default.asp>

- <input type="text"> (default value)
- <input type="button">
- <input type="checkbox">
- <input type="color">
- <input type="date">
- <input type="datetime-local">
- <input type="email">
- <input type="file">
- <input type="hidden">
- <input type="image">
- <input type="month">

Please Join in below link

Instagram: https://www.instagram.com/rba_infotech/

LinkedIn: <https://www.linkedin.com/company/rba-infotech/>

Facebook:

<https://www.facebook.com/people/RBA-Infotech/100043552406579/>

- <input type="number">
- <input type="password">
- <input type="radio">
- <input type="range">
- <input type="reset">
- <input type="search">
- <input type="submit">
- <input type="tel">
- <input type="time">
- <input type="url">
- <input type="week">

Ex:

http://register.rediff.com/register/register.php?FormName=user_details

<https://ssc.nic.in/Registration/Home>

https://www.jeevansathi.com/register/customreg/15?source=d_b_genlp1&sh1=Meet%20over&sh2=10%20lakh%20profiles&campaignid=669875208&adgroupid=36667840951&macthtype=e&device=c&keyword=jeevansathi&qclid=CjwKCAjw-sqKBhBjEiwAVaQ9a3VGwgE648WCwwIF_uBpj49HpXIYP2d40yVVScURnubnNhywJk5ZyxoC8QcQAvD_BwE

<https://www.naukri.com/registration/createAccount?othersrcp=22636>

HTML Attributes:

https://www.tutorialspoint.com/html/html_attributes.htm

https://www.w3schools.com/html/html_attributes.asp

2. Locators

There are **8 ways** to locate a WebElements

Please Join in below link

Instagram: https://www.instagram.com/rba_infotech/

LinkedIn: <https://www.linkedin.com/company/rba-infotech/>

Facebook:

<https://www.facebook.com/people/RBA-Infotech/100043552406579/>

- 1. ID**
- 2. Name**
- 3. Class (className)**
4. Xpath
5. CSS Selector
6. Tag Name - driver.findElements(By.tagName("a"))
7. Link Text – Anchor Link
8. Partial Link Text

3. Preference Guidelines:

- **First Choice:** Use ID for unique identification.
- **Second Choice:** Use Name if ID is not available.
- **Third Choice:** Use CSS Selectors for their speed and flexibility.
- **Fourth Choice:** Use XPath for complex queries or when elements are not easily identifiable by other locators.
- **Avoid:** Relying on Class Name or Tag Name alone, as they may not uniquely identify elements.

4. Xpath

Xpath is used when there are no other locators such as id, name, class name.

1.Two Types of Xpath

1. **Absolute Xpath** - starts with / and from root node /html - if there is a change of element in webpage, which leads the change of absolute xpath
2. **Relative Xpath** - starts with // - if forms the xpath from the previous nearby element which has id . Even if there is change of element in website most probably which will not make any change in the relative xpath possibility of relative xpath's working is very high.

2.When to use Absolute Xpath

Relative xpath cannot be used if the id of the xpath is keep changing.

Ex.

Please Join in below link

Instagram: https://www.instagram.com/rba_infotech/

LinkedIn: <https://www.linkedin.com/company/rba-infotech/>

Facebook:

<https://www.facebook.com/people/RBA-Infotech/100043552406579/>

In Facebook, **id** of the Create New Account is **keep changing** every time page is launched, this can be located by only Absolute Xpath.

3.Xpath functions (or) Customized Xpath (or) Dynamic Xpath

- **Xpath with Attribute** -> //<element>[@class='mbs _52lq fsl fwb fcb'] (in facebook account creation)
- **Contains()** – look for full or partial attribute value
 - o //<any element>[contains(@name,'first')]
- **Start-with()**
 - o //*[starts-with(@id,'any starting text here')]
 - o Ex
 - Id = u_b_5 (infy.com, name textbox)
 - Id=u_b_8
 - o Starts-with(@id,'u_b')
- **ends-with()**
- **Text()** – exact match with the text which is **between the open and closed elements / tags**
 - o //*[text()='Request OTP']
 - o
- **Index (Position())**
 - o //*[@id='anytext']/div/a/div[1]/div[position()=3]/p
 - o //*[@id='anytext']/div/a/div[1]/div[position() >= 10 and position() <=15]/p

Or

 - o //*[@id='anytext']/div/a/div[1]/div[3]/p
- **And & OR operations**

Contains()

Syntax:

Please Join in below link

Instagram: https://www.instagram.com/rba_infotech/

LinkedIn: <https://www.linkedin.com/company/rba-infotech/>

Facebook:

<https://www.facebook.com/people/RBA-Infotech/100043552406579/>

```
//*[contains(attribute, ' ')]  
//*[contains(text(),'sample text')]
```

4. Xpath Axes

Axes means relationships of current context node

Purpose:

Dynamic Xpath

To get all the elements with the Xpath

parent	child
ancestor	descendant
ancestor-or-self	descendant-or-self
preceding	following
preceding-sibling	following-sibling

```
//select[@name="birthday_year"]/parent::div  
//select[@name="birthday_year"]/child::div  
//select[@name="birthday_year"]/ancestor::div  
//*[@id="reg_form_box"]/descendant::div  
//*[@id="password_field"]/ancestor::div[2]
```

The following is the example for Xpath Axes in **Amazon** website for **laptop** search

```
//*[@id="search"]/div[1]/div[1]/div/span[3]/div[2]/div[5]/following::div  
//*[@id="search"]/div[1]/div[1]/div/span[3]/div[2]/div[7]/preceding::div
```

Please Join in below link

Instagram: https://www.instagram.com/rba_infotech/

LinkedIn: <https://www.linkedin.com/company/rba-infotech/>

Facebook:

<https://www.facebook.com/people/RBA-Infotech/100043552406579/>

5. CSS Selector

1. ID

- o #From - From is id
- o E.g: #id

2. Class

- o .btnSubmit - btnSubmit is class name

3. Attribute

- o Input[name='username']

4. Sub-string

- o To locate element whose value starting with - ^
In facebook email text box,
input[placeholder^='Email']
- o To locate element whose value ending with - \$
input[placeholder\$='number']
- o To locate element whose value contains (Sub-string) - *
*

5. Inner text

6. Wait (Test / Page Synchronization)

1. Types of Waits:

1. Implicit Wait
2. Explicit Wait
3. Fluent Wait
4. Other Waits

Please Join in below link

Instagram: https://www.instagram.com/rba_infotech/

LinkedIn: <https://www.linkedin.com/company/rba-infotech/>

Facebook:

<https://www.facebook.com/people/RBA-Infotech/100043552406579/>

- a. Thread.sleep()
- b. pageLoadTimeout();
- c. setScriptTimeout();

Few other waits:

- 1. Thread.sleep()
- 2. Pageloadtimeout()

This sets the time to wait for a page to load completely before throwing an error. If the timeout is negative, page loads can be indefinite.

- The pageLoadTimeout limits the time that the script allots for a web page to be displayed
- If the page loads within the time then the script continues, If the page does not load within the timeout the script will be stopped by a TimeoutException

Syntax:

```
driver.manage().timeouts().pageLoadTimeout(100, SECONDS);
```

- 3. setScriptTimeout()

This is used to set the amount of time the WebDriver must wait for an asynchronous script to finish execution before throwing an error. If the timeout is negative, then the script will be allowed to run indefinitely.

The default timeout for **setScriptTimeout** method is zero. If you do not set time, then there are chances that **executeAsyncScript** method may fail because the JavaScript code may take more than the time allotted. To avoid such failures, set the setScriptTimeout. This is mainly used for Javascript objects and executors.

Implicitly Wait -> DOM Structure -> JavaScript

E.g:

```
driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS)
```

Please Join in below link

Instagram: https://www.instagram.com/rba_infotech/

LinkedIn: <https://www.linkedin.com/company/rba-infotech/>

Facebook:

<https://www.facebook.com/people/RBA-Infotech/100043552406579/>

5. Explicit Wait -> 60, condition (profile image) – default – every sec

E.g:

```
WebDriverWait wait = new WebDriverWait(driver,50);
wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div[1]/div[2]/div/section[1]/div[1]/div[3]/div[2]/div/span")));
```

6. Fluent Wait 0 > 60, 5, condition (profile image)

Eg:

```
Wait<WebDriver> wait = new FluentWait<WebDriver>(driver)
    .withTimeout(20, TimeUnit.SECONDS)
    .pollingEvery(4, TimeUnit.SECONDS)
    .ignoring(NoSuchElementException.class);
```

<http://www.worldslongestwebsite.com/>

<https://alaskatrips.poweredbygps.com/>

<https://www.softwaretestingmaterial.com>

2. Implicit Wait:

Implicit Wait - It instructs the web driver to wait for some time by poll the DOM. Once you declared **implicit wait it will be available for the entire life of web driver instance until driver life span.** By default the value **will be 0**. If you set a longer default, then the behavior will poll the DOM on a periodic basis depending on the browser/driver implementation.

Throws a NoSuchElementException when the element is **not present** in the DOM

Please Join in below link

Instagram: https://www.instagram.com/rba_infotech/

LinkedIn: <https://www.linkedin.com/company/rba-infotech/>

Facebook:

<https://www.facebook.com/people/RBA-Infotech/100043552406579/>

Throws a ElementNotVisibleException when element is present in the DOM, however, it is hidden and cannot be interacted with

✓

3. Explicit Wait:

- ✓ Explicit wait in Selenium waits until the specified condition matches. It throws **only TimeoutException, will never throw NoSuchElementException** depending on your expected condition

4. Fluent Wait:

The Fluent Wait command defines the maximum amount of time for Selenium WebDriver to wait for a certain condition to appear. It also defines the frequency with which WebDriver will check if the condition appears before throwing the “ElementNotVisibleException”.

To put it simply, Fluent Wait looks for a web element repeatedly at regular intervals until timeout happens or until the object is found.

Fluent Wait commands are most useful when interacting with web elements that can sometimes take more time than usual to load. This is largely something that occurs in Ajax applications.

While using Fluent Wait, it is possible to set a default polling period as needed. The user can configure the wait to ignore any exceptions during the polling period.

```
Wait wait = new FluentWait(driver)
    .withTimeout(30, SECONDS)
    .pollingEvery(5, SECONDS)
    .ignoring(NoSuchElementException.class);
```

Please Join in below link

Instagram: https://www.instagram.com/rba_infotech/

LinkedIn: <https://www.linkedin.com/company/rba-infotech/>

Facebook:

<https://www.facebook.com/people/RBA-Infotech/100043552406579/>

```

WebElement foo = wait.until(new Function() {
    public WebElement apply(WebDriver driver) {
        return driver.findElement(By.id("foo"));
    }
});
```

Note: Implicit, Explicit and Fluent waits, are **dynamic waits**. What are dynamic waits? Consider a situation where you have given TimeOut value as 20 seconds. If the element is loaded in 5 seconds, then **rest 15 seconds will be ignored**. It won't wait till Timeout value is completed i.e 20 seconds. That's why all waits are considered as dynamic waits.

<https://www.swtestacademy.com/selenium-webdriver-wait/>

Solutions: We always get confuse when it comes to using Wait commands, to better understand it we need to remember that there is a difference between several scenarios:

An **element not being present at all** in the DOM.

An **element being present in the DOM but not visible**. (Ex. image and Video loading)

An **element being present in the DOM but not enabled**. (i.e. In the form, submit button will be enabled only after filling mandatory fields)

There are pages which get displayed with the JavaScript, the elements are already **present in the browser DOM, but are not visible**. **The implicit wait only waits for an element to appear in the DOM, so it returns immediately**, but when you try to interact with the element you get a NoSuchElementException. You could test this hypothesis by writing a helper method that explicit wait for an element to be visible or clickable.

Implicit Wait	Explicit Wait
---------------	---------------

Please Join in below link

Instagram: https://www.instagram.com/rba_infotech/

LinkedIn: <https://www.linkedin.com/company/rba-infotech/>

Facebook:

<https://www.facebook.com/people/RBA-Infotech/100043552406579/>

<ul style="list-style-type: none"> Implicit Wait time is applied to globally for the life of the WebDriver instance. It means that the driver will wait for the specified time before throwing a NoSuchElementException. 	<ul style="list-style-type: none"> Explicit Wait time is applied only to those elements which are intended by us. This can be repeated whenever necessary.
<ul style="list-style-type: none"> In Implicit Wait, we need not specify “ExpectedConditions” on the element to be located 	<ul style="list-style-type: none"> In Explicit Wait, we need to specify “ExpectedConditions” on the element to be located
<ul style="list-style-type: none"> It is recommended to use when the elements are located with the time frame specified in Selenium implicit wait 	<ul style="list-style-type: none"> It is recommended to use when the elements are taking long time to load and also for verifying the property of the element like(visibilityOfElementLocated, elementToBeClickable,elementToBeSelected)

What determines which xpath to be chosen

If the id of the Relative xpath is dynamic, you cannot use relative xpath, in such a situation, you can use Absolute Xpath

<https://www.easemytrip.com/>

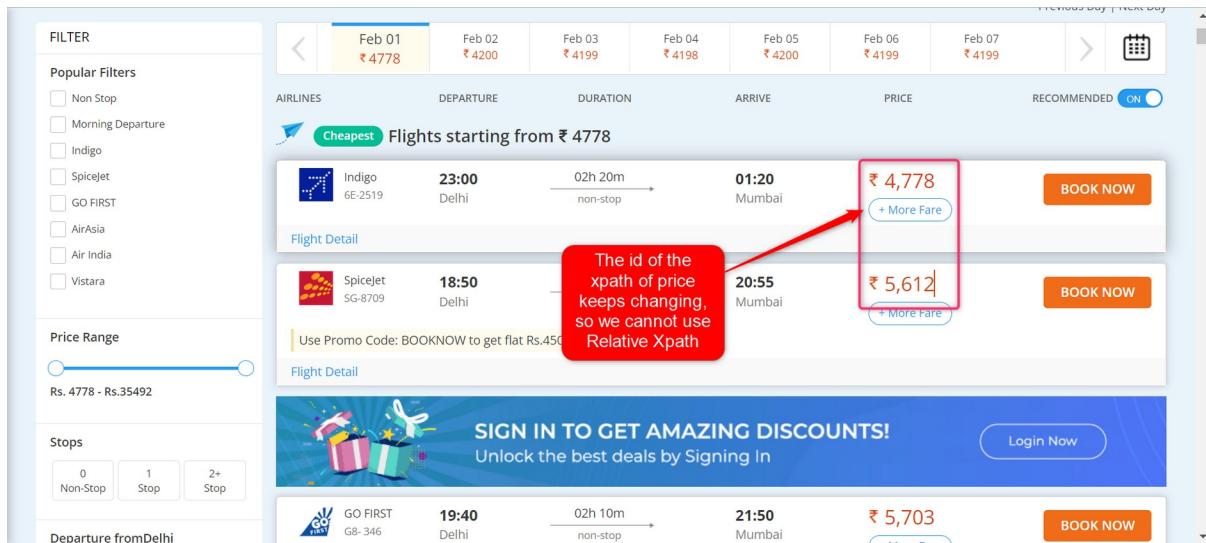
Please Join in below link

Instagram: https://www.instagram.com/rba_infotech/

LinkedIn: <https://www.linkedin.com/company/rba-infotech/>

Facebook:

<https://www.facebook.com/people/RBA-Infotech/100043552406579/>



7. Browser Navigation Commands

1. `navigate().to("www.amazon.com")`
2. `navigate.back()`
3. `navigate.forward()`
4. `navigate.refresh()`

8. Visibility of WebElement

1. `isDisplayed()`

2. `isEnabled()`

3. `isSelected()`

Please Join in below link

Instagram: https://www.instagram.com/rba_infotech/

LinkedIn: <https://www.linkedin.com/company/rba-infotech/>

Facebook:

<https://www.facebook.com/people/RBA-Infotech/100043552406579/>

9. Window Handles

1. getWindowHandle – return unique string parent window
2. getWindowHandles – returns set of strings of parent and child windows.

10. Web Table

How to retrieve the data from table

<https://money.rediff.com/gainers/bsc/daily/groupa>

11. Actions Class (Keyboard and Mouse activities)

1. Action Class

1. Actions Class

Actions class is based on **builder design pattern** which builds a composite action.

2. Keyboard Activities

- a. Alt
- b. Ctrl
- c. Shift
- d. Enter
- e. UpArrow, DownArrow, LeftArrow and Right Arrow
- f. Function Keys

3. Mouse Activities / Events

- g. Click
- h. Right Click (contextClick())
- i. moveToElement
- j. Double Click
- k. Drag and Drop
- l. Click and Hold
- m. Move By Offset

Please Join in below link

Instagram: https://www.instagram.com/rba_infotech/

LinkedIn: <https://www.linkedin.com/company/rba-infotech/>

Facebook:

<https://www.facebook.com/people/RBA-Infotech/100043552406579/>

Keyboard Actions:

Resources: <https://www.toolsqa.com/selenium-webdriver/keyboard-events-in-selenium/>

Practise: <https://demoqa.com/text-box>

Below is the code inn normal way

```
driver.get("https://demoqa.com/text-box");

// Enter the Full Name
WebElement fullName =
driver.findElement(By.id("userN"));
fullName.sendKeys("Mr.Peter Haynes");

//Enter the Email
WebElement email=driver.findElement(By.id("userEmail"));
email.sendKeys("PeterHaynes@toolsqa.com");

// Enter the Current Address
WebElement
currentAddress=driver.findElement(By.id("currentAddress"));
currentAddress.sendKeys("43 School Lane London EC71 9GO");

// Copy the Current Address
currentAddress.sendKeys(Keys.CONTROL);
currentAddress.sendKeys("A");
currentAddress.sendKeys(Keys.CONTROL);
currentAddress.sendKeys("C");

//Press the TAB Key to Switch Focus to Permanent Address
currentAddress.sendKeys(Keys.TAB);

//Paste the Address in the Permanent Address field
WebElement
permanentAddress=driver.findElement(By.id("permanentAddress"));
permanentAddress.sendKeys(Keys.CONTROL);
permanentAddress.sendKeys("V");
```

Please Join in below link

Instagram: https://www.instagram.com/rba_infotech/

LinkedIn: <https://www.linkedin.com/company/rba-infotech/>

Facebook:

<https://www.facebook.com/people/RBA-Infotech/100043552406579/>

Methods Available in Selenium Actions Class:

Keyboard Events Using Selenium Actions Class API:

The Keyboard interface has the below mentioned methods:

- sendKeys(keysToSend) : sends a series of keystrokes onto the element
- keyDown(theKey) : Sends a key press without release it. Subsequent actions may assume it as pressed. (example: Keys.ALT, Keys.SHIFT, or Keys.CONTROL)
- keyUp(theKey): Performs a key release

Below is the code which uses Keyboard activities:

```
driver.get("https://demoqa.com/text-box");

//Create object of the Actions class
Actions actions = new Actions(driver);

// Enter the Full Name
WebElement fullName =
driver.findElement(By.id("userNmae"));
fullName.sendKeys("Mr.Peter Haynes");

//Enter the Email
WebElement email=driver.findElement(By.id("userEmail"));
email.sendKeys("PeterHaynes@toolsqa.com");

// Enter the Current Address
WebElement
currentAddress=driver.findElement(By.id("currentAddress"));

currentAddress.sendKeys("43 School Lane London EC71 9GO");

// Select the Current Address using CTRL + A
actions.keyDown(Keys.CONTROL);
```

Please Join in below link

Instagram: https://www.instagram.com/rba_infotech/

LinkedIn: <https://www.linkedin.com/company/rba-infotech/>

Facebook:

<https://www.facebook.com/people/RBA-Infotech/100043552406579/>

```

actions.sendKeys("a");
actions.keyUp(Keys.CONTROL);
actions.build().perform();

// Copy the Current Address using CTRL + C
actions.keyDown(Keys.CONTROL);
actions.sendKeys("c");
actions.keyUp(Keys.CONTROL);
actions.build().perform();

//Press the TAB Key to Switch Focus to Permanent Address
actions.sendKeys(Keys.TAB);
actions.build().perform();

//Paste the Address in the Permanent Address field using CTRL
+ V
actions.keyDown(Keys.CONTROL);
actions.sendKeys("v");
actions.keyUp(Keys.CONTROL);
actions.build().perform();

```

Mouse Movements

Mouse Events Using Selenium Actions Class API:

- click(): Simply click on element
- doubleClick (): Double clicks onElement
- contextClick() : Performs a context-click (right click) on an element
- clickAndHold(): Clicks at the present mouse location (without releasing)
- dragAndDrop(source, target): Invokes click-and-hold at the source location and moves to the location of the target element before releasing the mouse. source – element to grab, target – element to release
- dragAndDropBy(source, xOffset, yOffset) : Performs click-and-hold at the source location, shifts by a given offset, then frees the mouse. xOffset – to shift horizontally, yOffset – to shift vertically
- moveByOffset(x-offset, y-offset): Shifts the mouse from its current position (or 0,0) by the given offset. x-offset – Sets the horizontal

Please Join in below link

Instagram: https://www.instagram.com/rba_infotech/

LinkedIn: <https://www.linkedin.com/company/rba-infotech/>

Facebook:

<https://www.facebook.com/people/RBA-Infotech/100043552406579/>

offset (negative value - shifting the mouse to the left), y-offset - Sets the vertical offset (negative value - shifting the mouse to the up)

- `moveToElement(toElement)`: It shifts the mouse to the center of the element
- `release()`: Releases the depressed left mouse button at the existing mouse location

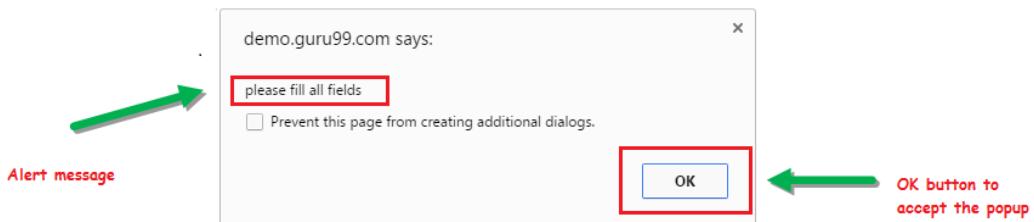
12. Alerts Windows

1.3 Types of Alerts

1. Simple Alerts - OK button
2. Confirmation Alert - Ok and Cancel button
3. Prompt Alerts - Text box which will accept some texts and have ok and cancel button

2. Simple Alert

The simple alert class in Selenium displays some information or warning on the screen.



3. Confirmation Alert.

This confirmation alert asks permission to do some type of operation.

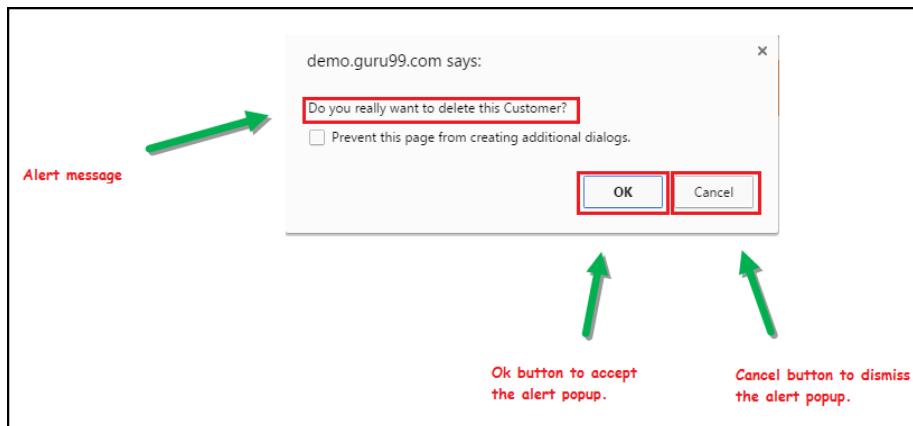
Please Join in below link

Instagram: https://www.instagram.com/rba_infotech/

LinkedIn: <https://www.linkedin.com/company/rba-infotech/>

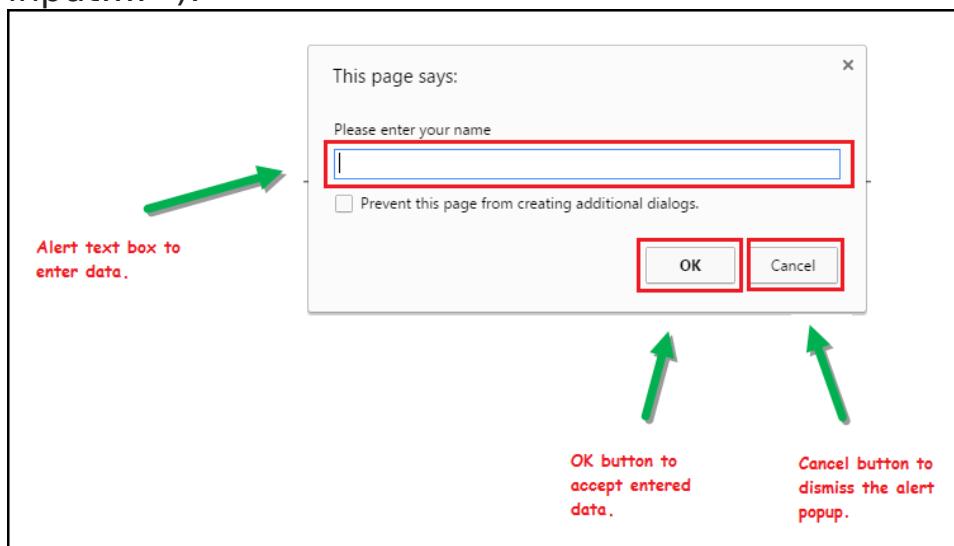
Facebook:

<https://www.facebook.com/people/RBA-Infotech/100043552406579/>



4. Prompt Alert

This Prompt Alert asks some input from the user and Selenium webdriver can enter the text using sendkeys("input....").



Accept() - to click "ok" button

Dismiss() - to click "cancel" button

getText() - to get Alert msg

sendKeys() - to type text

Please Join in below link

Instagram: https://www.instagram.com/rba_infotech/

LinkedIn: <https://www.linkedin.com/company/rba-infotech/>

Facebook:

<https://www.facebook.com/people/RBA-Infotech/100043552406579/>

13. JavaScriptExecutor

What is JavaScriptExecutor?

JavaScriptExecutor is an Interface that helps to execute [JavaScript](#) through Selenium Webdriver. JavaScriptExecutor provides two methods "executescript" & "executeAsyncScript" to run javascript on the selected window or current page.

Why do we need JavaScriptExecutor?

In Selenium Webdriver, locators like XPath, CSS, etc. are used to identify and perform operations on a web page.

In case, these locators do not work you can use JavaScriptExecutor. You can use JavaScriptExecutor to perform an desired operation on a web element.

Two Methods:

1. executeAsyncScript – Asynchronous calls
2. executeScript

Syntax:

```
JavascriptExecutor js = (JavascriptExecutor) driver;  
js.executeScript( Script, Arguments );
```

executeAsyncScript:

1. Performing a sleep in the browser under test.

executeScript:

1. Generate Alert Window

Please Join in below link

Instagram: https://www.instagram.com/rba_infotech/

LinkedIn: <https://www.linkedin.com/company/rba-infotech/>

Facebook:

<https://www.facebook.com/people/RBA-Infotech/100043552406579/>

2. Capture Scrape Data and Navigate to different pages
3. Scroll Down the webpage

<https://www.swtestacademy.com/selenium-javascriptexecutor/>

14. Iframes

Iframe is nothing but Inline Frame (webpage) which can be embedded into another webpage. In general ads displayed in the websites are the examples of Iframe.

AJAX – Asynchronous Javascript And XML

Why?

The Iframes normally can not be located or identified like locating other Web Elements.

How to switch over the elements in iframes using Web Driver commands:

Basically, we can switch over the elements in frames using 3 ways.

- By Index
- By Name or Id
- By Web Element

Please Join in below link

Instagram: https://www.instagram.com/rba_infotech/

LinkedIn: <https://www.linkedin.com/company/rba-infotech/>

Facebook:

<https://www.facebook.com/people/RBA-Infotech/100043552406579/>

15. Model Window

<http://uitestpractice.com/>

When a **modal** window opens, it opens inside the current page. ...
With **popup** windows, users can often miss them because they don't grab the user's attention. When users click the browser window, the browser window comes to the front and the **popup** window hides behind it.

16. Exceptions in Selenium

1. **WebDriverException**: This is the base exception class for all WebDriver-related exceptions.
2. **NoSuchElementException**: Thrown when an element could not be found on the page.
3. **TimeoutException**: Thrown when a command does not complete in the expected time.
4. **ElementNotVisibleException**: Thrown when an element is present in the DOM but is not visible on the page.
5. **ElementNotSelectableException**: Thrown when an element is present in the DOM but is not selectable.
6. **StaleElementReferenceException**: Thrown when the element reference is no longer valid, usually because the DOM has been refreshed.
7. **InvalidElementStateException**: Thrown when attempting to perform an action on an element that is in an invalid state.
8. **InvalidSelectorException**: Thrown when the selector used to locate an element is invalid.
9. **NoSuchFrameException**: Thrown when the frame specified by the locator cannot be found.
10. **NoSuchWindowException**: Thrown when the window specified by the handle cannot be found.
11. **UnhandledAlertException**: Thrown when an unhandled alert is present.
12. **UnexpectedAlertPresentException**: Thrown when an unexpected alert is present.

Please Join in below link

Instagram: https://www.instagram.com/rba_infotech/

LinkedIn: <https://www.linkedin.com/company/rba-infotech/>

Facebook:

<https://www.facebook.com/people/RBA-Infotech/100043552406579/>

13. **InvalidCookieDomainException**: Thrown when attempting to add a cookie under a different domain than the current page.
14. **InvalidCoordinatesException**: Thrown when attempting to move the mouse to an invalid coordinate.
15. **SessionNotCreatedException**: Thrown when a new session could not be created.
16. **ElementClickInterceptedException**: Thrown when an element is clicked, but another element obscures it.
17. **ElementNotInteractableException**: Thrown when an element is present in the DOM but cannot be interacted with.
18. **MoveTargetOutOfBoundsException**: Thrown when attempting to move the mouse to a target that is outside the browser's viewport.

17. Turn off notifications

```
ChromeOptions ops = new ChromeOptions();
ops.addArguments("--disable-notifications");
driver = new ChromeDriver(ops);
```

18. Take Screen Shot

```
//Convert web driver object to TakeScreenshot

TakesScreenshot scrShot =((TakesScreenshot)webdriver);

//Call getScreenshotAs method to create image file

File SrcFile=scrShot.getScreenshotAs(OutputType.FILE);

//Move image file to new destination
```

Please Join in below link

Instagram: https://www.instagram.com/rba_infotech/

LinkedIn: <https://www.linkedin.com/company/rba-infotech/>

Facebook:

<https://www.facebook.com/people/RBA-Infotech/100043552406579/>

```
File DestFile=new File(filePath);

//Copy file at destination

FileUtils.copyFile(SrcFile, DestFile);
```

19. Advanced Selenium

20. AutoIT

What is AutoIT?

AutoIT is a freeware scripting language designed for automating windows GUI. It uses a combination of mouse movement, keystrokes and window control manipulation to automate a task which is not possible by selenium webdriver.

Why use AutoIT?

[Selenium](#) is an open source tool that is designed to automate web-based applications on different browsers but to handle window GUI and non HTML popups in ap

To handle below elements

Windows authentication box

File uploading dialog box

Any other non-browser interaction

21. Robot Class

What is a Robot class?

Time plays a major role while testing an application and we need to make sure we complete the desired task within the specific time.

Please Join in below link

Instagram: https://www.instagram.com/rba_infotech/

LinkedIn: <https://www.linkedin.com/company/rba-infotech/>

Facebook:

<https://www.facebook.com/people/RBA-Infotech/100043552406579/>

A *Robot class* in **Selenium** is used to generate native system input events for test automation, self-running demos, and other applications where you need control over the mouse and keyboard. **WebDriver** cannot handle the OS popups, so in Java 1.3, Robot class was introduced.

The primary purpose of this Robot class is to facilitate **automated testing** for Java platform implementations. In simple terms, I would say this class provides control over the mouse and keyboard devices.

This is very easy to implement as it can be easily integrated with current automation framework

Now, let us understand the importance of this class

Importance of Robot class

- Uploading a file is easy when we use the robot class
- It can simulate and handle the mouse and keyboard functions
- It can handle pop-ups as well

22. How to execute Selenium Script from Command Line

Please Join in below link

Instagram: https://www.instagram.com/rba_infotech/

LinkedIn: <https://www.linkedin.com/company/rba-infotech/>

Facebook:

<https://www.facebook.com/people/RBA-Infotech/100043552406579/>